Custom Git Repository Feature Overview
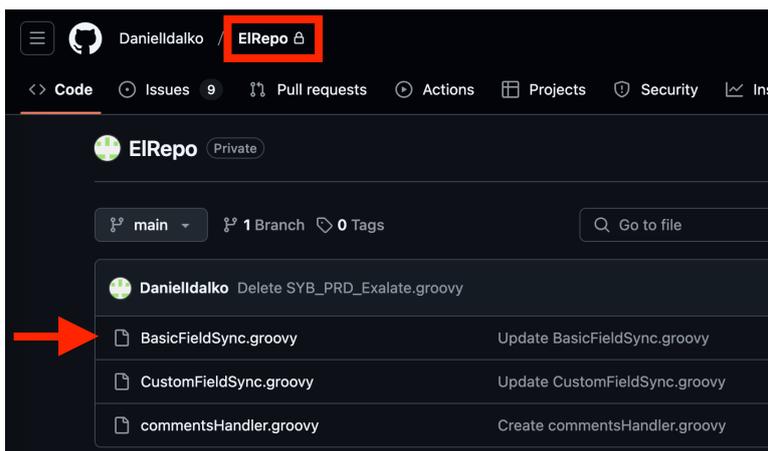
Last Modified on 01/15/2026 2:02 pm EST

As your Exalate integration expands, you may need a centralized version control solution for your scripts. You can do that now in accordance with industry standards.
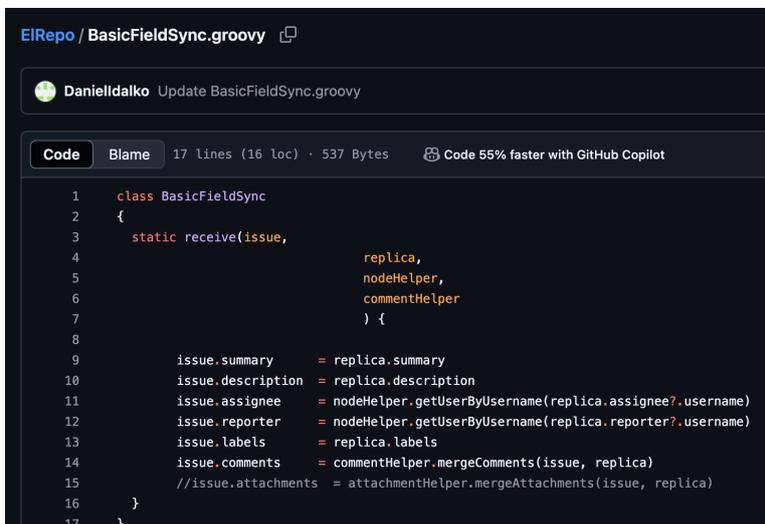
Custom Git Repository Feature Overview

GitHub is a platform for code version control and deployment widely used by developers all around the globe. Since Exalate uses a script-based console, you might want to use Git to control your Exalate scripts' versions.

Exalate provides a solution to integrate GitHub repositories into the Exalate console by using "External scripts". This is basically a library that you can refer to inside your Groovy script and import classes from.

Here is an example of a GitHub repository(ElRepo) which contains a script that you can use in the Exalate console(BasicFieldSync.groovy):

This is how the class definition looks in that script:

Given that the node has been correctly configured to access this repository(see notes below), you can then reference it inside the Exalate console as follows:

Once this is implemented, you can now make changes to the script rules directly in the Git repository Class script instead of accessing the Exalate console, and enjoy the flexibility that comes with it, along with the version control and other Git features.
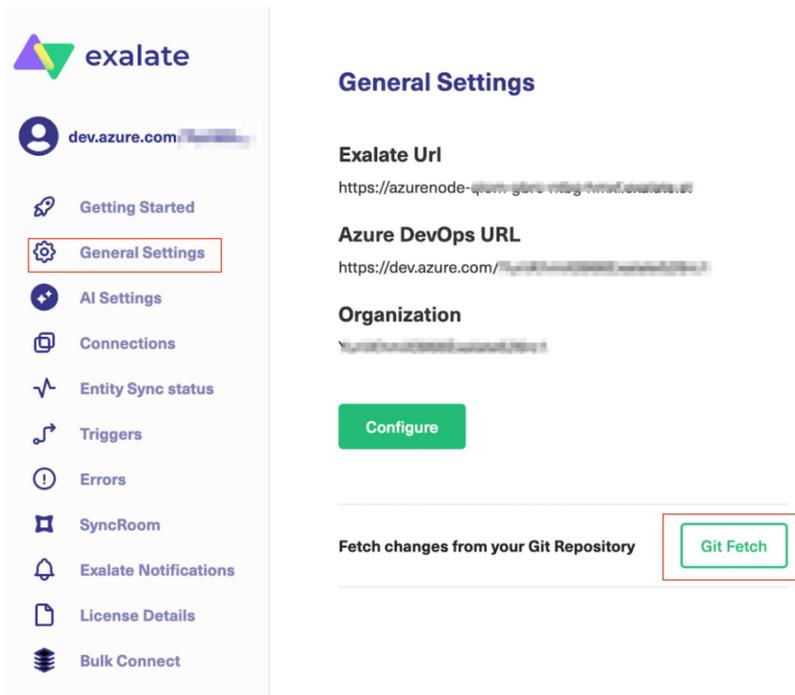
Important Notes

- There's a 15-minute update time between changes performed in the Git repository to be reflected on the Exalate console itself.

- Configuration of a custom Git repo is performed by our cloud team on each node separately.

- It's possible to configure the node to fetch from private repositories.

- Example scripts and more information on the scripting side of the action can be found here and here.

Manual Git Fetch

For situations where you need changes to be reflected immediately without waiting for the automatic fetch cycle, Exalate now provides the capability to trigger a manual Git fetch directly from the Exalate console:

- Navigate to **General Settings** in your Exalate console.
- Perform the **"Git Fetch"** action

After pressing **Git Fetch**, changes will be applied immediately without requiring a node restart.